

BAB 3

METODE PENELITIAN

Penelitian ini merupakan jenis penelitian rekayasa atau rancang bangun. Kegiatan penelitian dimulai dari identifikasi permasalahan, pemetaan proses, penelusuran akar masalah, hingga perancangan dan pengembangan sistem yang ditujukan untuk mengatasi atau meminimalkan permasalahan yang ada. Uraian mengenai bahan, alat, metode pengembangan, serta tahapan penelitian disajikan sebagai dasar dalam merancang *platform* literasi berbasis *website* yang dilengkapi dengan sistem rekomendasi yang memanfaatkan algoritma *content-based filtering*.

3.1 BAHAN DAN ALAT PENELITIAN

Penelitian ini mengumpulkan data dan informasi dari beragam referensi. Sumber utama berasal dari literatur yang sesuai dan berkaitan, seperti jurnal dan hasil penelitian sebelumnya, yang berkaitan dengan pengembangan sistem rekomendasi berbasis algoritma *content-based filtering* serta perancangan *platform* literasi berbasis web.

Sementara itu, data untuk sistem rekomendasi diperoleh melalui proses *web scraping* dari situs resmi gramedia.com menggunakan *chrome extension* bernama Easy Scraper, mencakup 500 data buku yang terdiri atas *cover*, judul, dan nama penulis. Adapun deskripsi buku dilengkapi secara manual melalui hasil pencarian di Google maupun dari laman resmi Gramedia. Data ini menjadi dasar penerapan algoritma *content-based filtering* dalam pengembangan sistem rekomendasi pada *platform* literasi berbasis *website*.

Sistem operasi dan aplikasi-aplikasi yang dipakai dalam pengembangan *platform* ini mencakup:

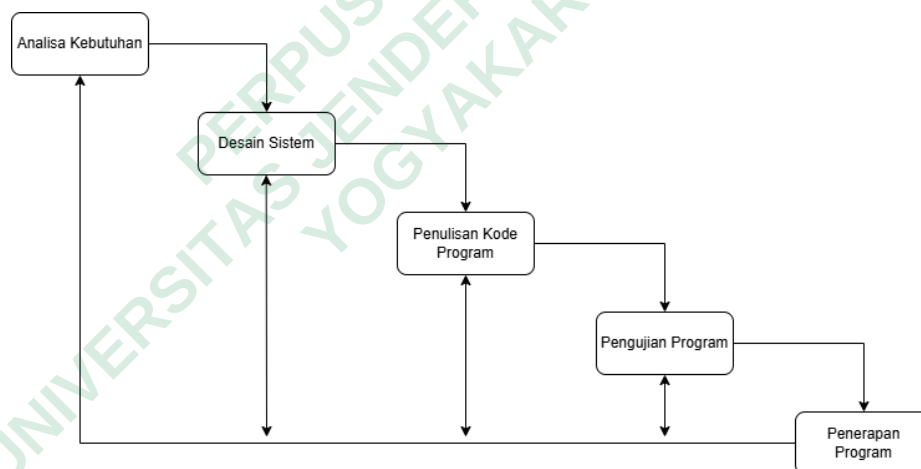
1. Windows 10
2. Python versi 3.11.0
3. PostgreSQL versi 7.3 atau lebih baru
4. Google Chrome
5. Visual Studio Code versi 1.96.3

6. Google Colab

7. Figma

3.2 JALAN PENELITIAN

Dalam proses perancangan dan pengembangan sistem, diperlukan metode pengembangan yang sistematis agar setiap tahapannya dapat berjalan dengan baik dan terstruktur. Penelitian ini menerapkan pendekatan *Waterfall* sebagai salah satu metode yang digunakan. Metode ini adalah model pengembangan perangkat lunak yang dilakukan secara bertahap dan berurutan, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya, layaknya aliran air terjun. Tahapan tersebut mencakup analisa kebutuhan, perancangan sistem, penulisan kode, pengujian, dan penerapan program (Mirajdandi et al., 2021).. Pendekatan ini dipilih karena sesuai untuk proyek dengan ruang lingkup yang jelas dan kebutuhan yang telah ditentukan sejak awal. Ilustrasi tentang metode ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Metode *Waterfall*

Secara garis besar, penelitian ini terbagi menjadi 4 tahap yaitu:

1. Analisa kebutuhan

Tahap ini mencakup analisis kebutuhan fungsional dan non-fungsional.

2. Desain sistem

Tahap desain sistem dalam metode *Waterfall* bertujuan menerjemahkan hasil analisis kebutuhan menjadi rancangan teknis sebagai pedoman pengembangan. Pada tahap ini dilaksanakan kegiatan berupa perancangan alur data dan proses menggunakan *use case diagram*, *activity diagram*, serta perancangan antarmuka pengguna (UI/UX) juga dirancang menggunakan Figma untuk memastikan tampilan sistem yang intuitif dan estetis. Seluruh hasil desain ini akan didokumentasikan sebagai acuan pada tahap implementasi.

3. Penulisan kode program

Pada tahap ini, penulisan kode program dilakukan dengan menggunakan bahasa pemrograman Python dengan bantuan *framework* Django, serta pembuatan basis data sesuai dengan perancangan yang telah dibuat sebelumnya, menggunakan *database server* PostgreSQL. Selain itu, dilakukan pula tahap *preprocessing* data untuk membersihkan dan menyesuaikan data buku sebelum dimasukkan ke dalam basis data, agar sesuai dengan format dan struktur yang dibutuhkan oleh sistem rekomendasi.

4. Pengujian program

Tahap keempat meliputi pengujian sistem yang telah dikembangkan dengan menerapkan metode *black box testing* yang bertujuan mengevaluasi fungsionalitas sistem yang dievaluasi berdasarkan data *input* dan *output* tanpa melihat proses internal yang terjadi. Serta pengujian kinerja sistem rekomendasi buku menggunakan *precision*.

5. Penerapan program

Tahap penerapan program tidak termasuk dalam penelitian ini karena ruang lingkup proyek dibatasi sampai pada tahap pengujian sistem guna memastikan bahwa fungsi-fungsi inti berjalan sesuai dengan kebutuhan yang telah ditetapkan.

3.3 ANALISA KEBUTUHAN

Tahap pertama dalam metode *Waterfall* adalah analisa kebutuhan, yaitu proses untuk menggali dan merumuskan apa saja yang dibutuhkan oleh sistem, baik secara fungsional maupun non-fungsional. Analisis ini dilakukan berdasarkan data dari situs Gramedia sebanyak 500 data buku yang mencakup judul, penulis, dan *cover*, yang selanjutnya digunakan untuk mendukung sistem rekomendasi berbasis *content-based filtering*. Penyusunan kebutuhan juga mengacu pada penelitian serupa dalam pengembangan *platform* literasi yang umumnya menekankan kemudahan akses konten, tampilan responsif, dan rekomendasi yang relevan.

3.3.1 Karakteristik Pengguna

Dalam sistem *platform* literasi ini, terdapat dua jenis pengguna yang akan memanfaatkan layanan tersebut, yaitu:

1. User

- User adalah masyarakat umum yang menggunakan sistem.
- User dapat melihat, melihat buku berdasarkan kategori, mencari, mendapatkan rekomendasi buku, dan mengunduh *file* buku.

2. Admin

- Admin adalah pihak pengelola sistem atau *website*.
- Admin dapat melihat data, mencari buku, membaca *file* buku, mengunduh *file* buku, menambah data buku baru, mengedit data yang sudah ada, serta menghapus data buku.

3.3.2 Kebutuhan Fungsional

1. Admin bisa melakukan *login* pada sistem
2. Admin dapat menambahkan data buku baru
3. Admin bisa melihat daftar buku
4. Admin dapat melakukan pencarian data buku
5. Admin dapat membaca *file* buku
6. Admin dapat mengunduh *file* buku
7. Admin mampu mengedit data buku yang sudah ada

8. Admin dapat menghapus data buku
9. Admin dapat melakukan *logout* dari sistem
10. *User* dapat melihat data buku
11. *User* dapat melihat data buku berdasarkan kategori
12. *User* dapat mencari judul buku dan penulis yang diinginkan
13. *User* dapat melihat hasil buku rekomendasi
14. *User* dapat membaca *file* buku
15. *User* dapat mengunduh *file* buku

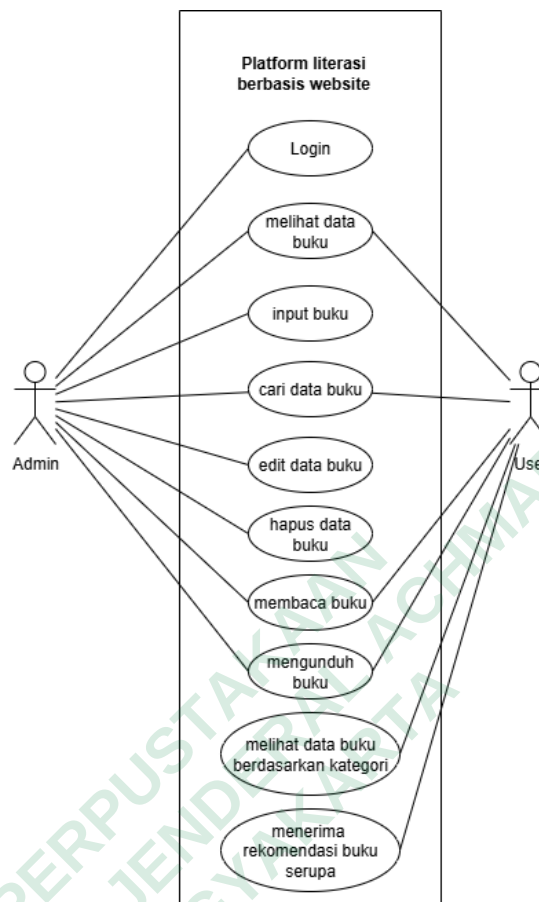
3.3.3 Kebutuhan Non-Fungsional

1. Sistem dapat diakses melalui peramban web seperti Google Chrome.
2. Sistem memiliki antarmuka yang dirancang agar intuitif dan mudah dipahami oleh pengguna.

3.4 DESAIN SISTEM

Tahap perancangan sistem dalam metode *Waterfall* merupakan proses menerjemahkan hasil analisis kebutuhan ke dalam bentuk desain teknis yang akan menjadi acuan pengembangan. Perancangan ini mencakup penyusunan alur data dan proses menggunakan *Unified Modeling Language* (UML), yang terdiri dari *use case diagram* dan *activity diagram*. Sementara itu, desain antarmuka pengguna (UI/UX) dibuat menggunakan Figma.

3.4.1 Use Case Diagram



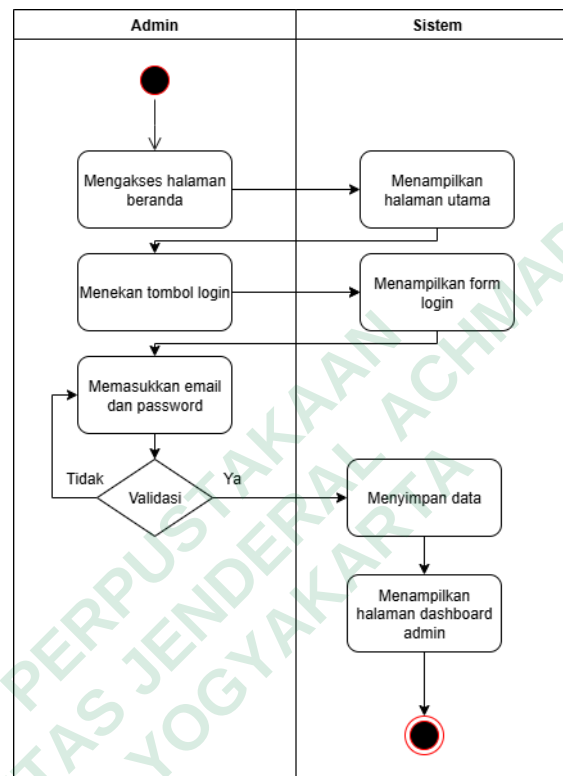
Gambar 3.2 Use case Diagram

Use case diagram digunakan untuk menggambarkan fungsionalitas yang diharapkan dari suatu sistem dengan menitikberatkan pada aktivitas yang dilaksanakan oleh sistem tersebut. Setiap *use case* menggambarkan interaksi antara aktor (pengguna atau entitas luar) dengan sistem saat melaksanakan fungsi tertentu. Dari *use case* dalam gambar 3.2. dapat ditarik kesimpulan bahwa *platform* literasi berbasis *website* memerlukan 2 aktor yaitu admin dan *user*.

3.4.2 Activity Diagram

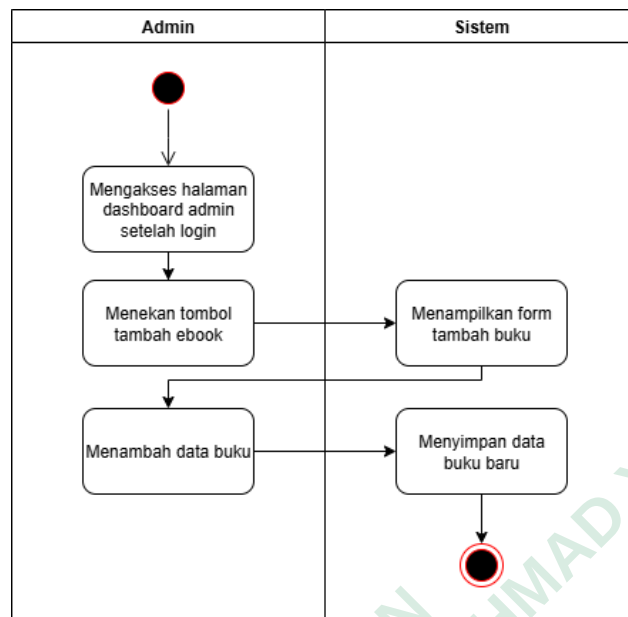
Pada *activity diagram login* admin, proses dimulai ketika admin mengakses *website*, sehingga sistem akan menampilkan halaman beranda. Selanjutnya, sistem menampilkan halaman utama yang terdapat tombol *login* maka sistem akan menampilkan formulir *login* yang harus diisi oleh admin dengan memasukkan

email dan *password*, kemudian menekan tombol *login*. Maka sistem akan memvalidasi data yang dimasukkan untuk memastikan keabsahan *email* dan kata sandi. Jika validasi berhasil, admin akan diarahkan ke halaman *dashboard* setelah berhasil *login*. *Activity diagram* untuk proses *login* ini ditampilkan pada Gambar 3.3.



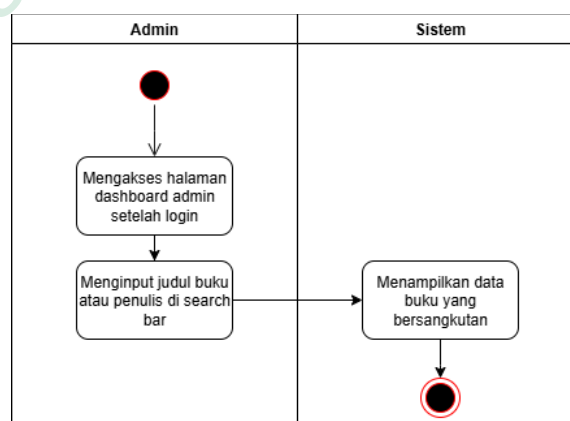
Gambar 3.3 Activity Diagram Login Admin

Pada *activity diagram* proses penambahan data buku, alur dimulai setelah admin berhasil masuk dan dialihkan ke halaman *dashboard* admin. Kemudian memilih tombol tambah *ebook* untuk menambahkan buku dan mengisi formulir yang telah disediakan dengan data buku yang dibutuhkan, seperti judul, penulis, genre, deskripsi, *cover* buku, dan *file* buku. Sesudah semua data diisi dengan lengkap, admin menekan tombol *submit*. sistem akan memproses *input* dan menyimpan data buku baru tersebut ke dalam basis data. *Activity diagram* untuk proses ini ditampilkan pada Gambar 3.4.



Gambar 3.4 Activity Diagram Tambah Buku

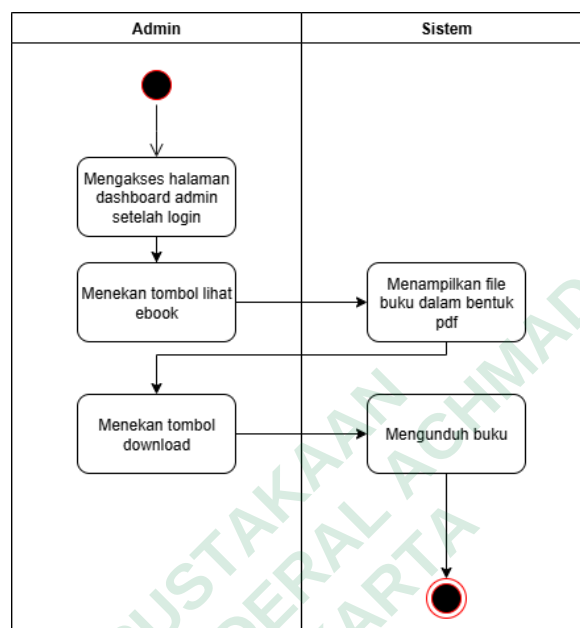
Pada *activity diagram* proses pencarian buku, alur dimulai ketika admin berhasil melakukan *login* dan diarahkan ke halaman *dashboard* admin. Pada halaman tersebut, terdapat fitur *search bar* yang terletak di sebelah kiri tombol *Logout*. Admin dapat memasukkan kata kunci berupa judul buku atau nama penulis ke dalam *search bar*. Setelah itu, sistem akan memproses *input* tersebut dan menampilkan hasil pencarian yang sesuai serta relevan dengan *keyword* yang dimasukkan. *Activity diagram* untuk proses ini ditampilkan pada Gambar 3.5.



Gambar 3.5 Activity Diagram Pencarian Buku Di Dashboard Admin

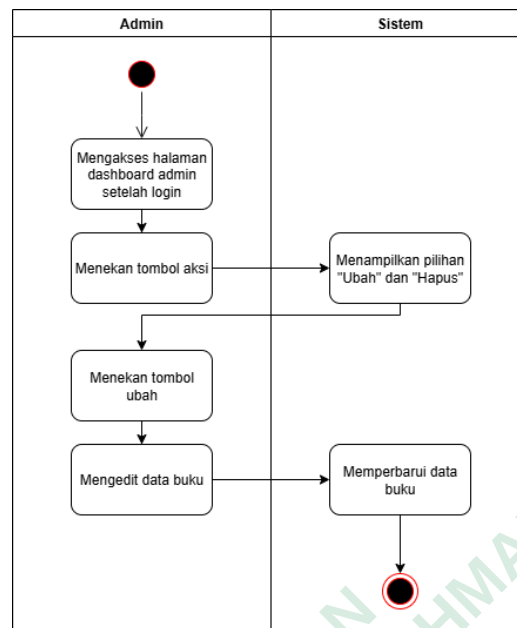
Pada *activity diagram* proses pencarian data buku, alur dimulai ketika admin berhasil masuk dan menuju ke halaman *dashboard* admin. Selanjutnya, *admin*

menekan tombol lihat ebook untuk melihat file buku yang tersedia. Sistem kemudian menampilkan file buku dalam bentuk pdf. Pada halaman tersebut, terdapat fitur *download* yang membuat admin dapat mengunduh buku tersebut. *Activity diagram* untuk proses ini dapat dilihat pada Gambar 3.6.



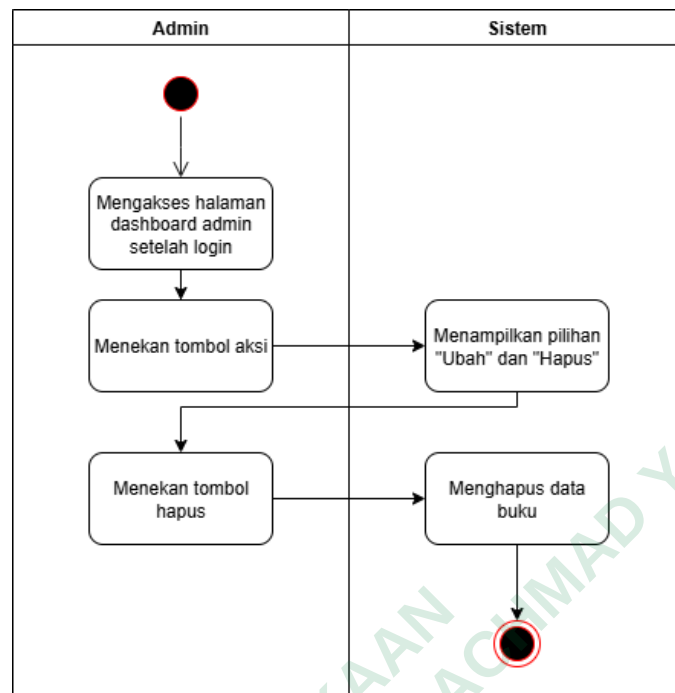
Gambar 3.6 Activity Diagram Membaca dan Mengunduh Buku oleh Admin

Pada *activity diagram* proses pengeditan data buku, alur dimulai setelah admin berhasil masuk dan diarahkan ke halaman *dashboard*. Admin kemudian menekan tombol aksi pada salah satu data buku yang ingin diubah, lalu memilih opsi *ubah* untuk melakukan pengeditan. Sistem akan menampilkan formulir yang telah terisi dengan data buku sebelumnya, dan admin dapat memperbarui informasi seperti judul, penulis, genre, deskripsi, *cover* buku, dan *file* buku. Setelah semua perubahan selesai dilakukan, admin menekan tombol *submit*. Sistem kemudian memproses *input* tersebut dan memperbarui data buku tersebut di dalam basis data. *Activity diagram* untuk proses ini ditampilkan pada Gambar 3.7.



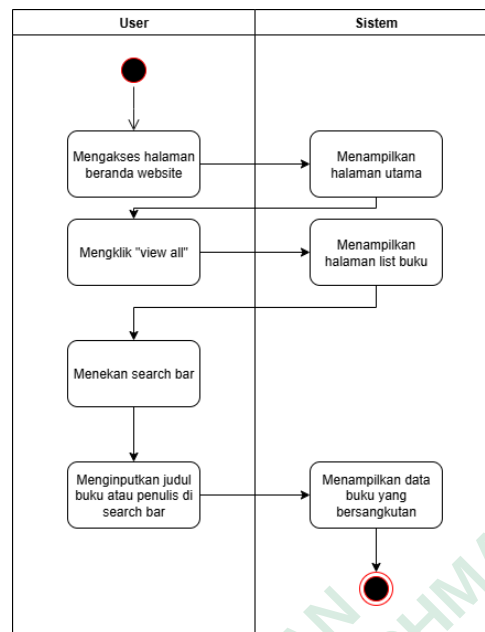
Gambar 3.7 Activity Diagram Edit Buku

Di *activity diagram* untuk proses penghapusan data buku, alur dimulai setelah admin berhasil masuk serta mengakses halaman *dashboard*. Selanjutnya admin menekan tombol aksi pada salah satu data buku yang ingin diubah, lalu memilih opsi hapus untuk melakukan penghapusan. Sistem kemudian akan menghapus data buku tersebut dari basis data. *Activity diagram* untuk proses ini ditampilkan pada Gambar 3.8.



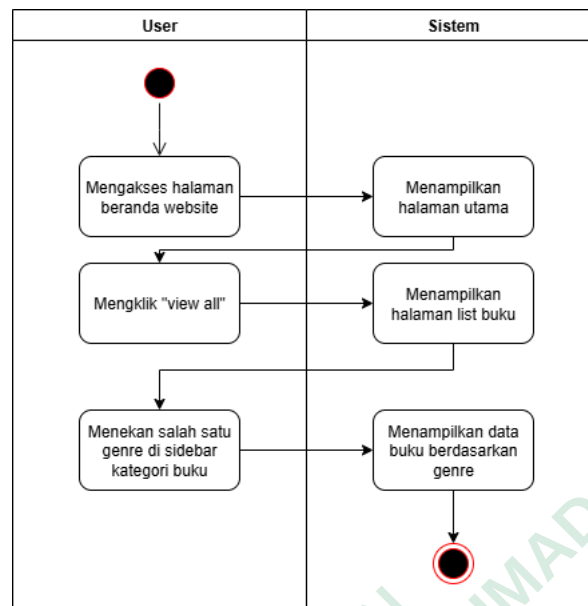
Gambar 3.8 Activity Diagram Hapus Buku

Pada *activity diagram* proses pencarian data buku, alur dimulai ketika *user* berada di halaman utama. Selanjutnya, *user* menekan tombol *View All* untuk melihat daftar buku yang tersedia. Sistem kemudian menampilkan halaman *list* buku. Pada halaman tersebut, terdapat fitur *search bar* yang terletak di sebelah kiri tombol *Logout*. Kata kunci berupa judul buku atau penulis dapat diinputkan ke dalam *search bar* oleh *user*. Setelah itu, sistem akan memproses *input* tersebut dan memberikan hasil pencarian buku yang relevan serta dengan *keyword* yang diinputkan. *Activity diagram* untuk proses ini ditampilkan pada Gambar 3.9.



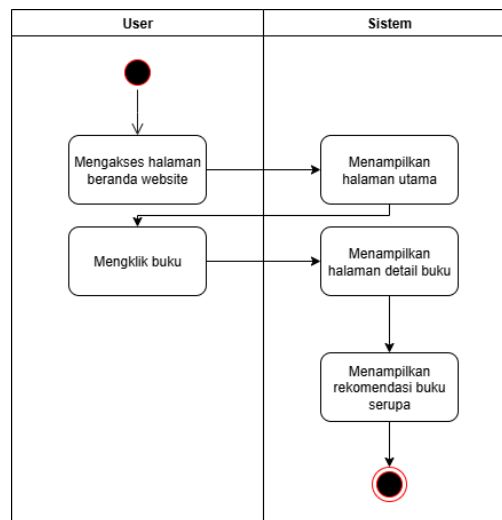
Gambar 3.9 Activity Diagram Pencarian Data Buku

Pada *activity diagram* proses melihat data buku berdasarkan kategori, alur dimulai ketika *user* berada di halaman utama. Setelah itu, *user* menekan tombol *View All* untuk mengakses daftar lengkap buku yang tersedia di sistem. Sistem kemudian menampilkan halaman *list* buku yang dilengkapi dengan *sidebar* di bagian kiri yang berisi berbagai pilihan genre atau kategori buku. *User* dapat memilih kategori yang diinginkan, lalu sistem akan menyaring dan menampilkan daftar buku sesuai genre tersebut. Proses ini memudahkan *user* dalam menelusuri koleksi buku sesuai dengan preferensi genre yang diminati. *Activity diagram* untuk proses ini ditampilkan pada Gambar 3.10.



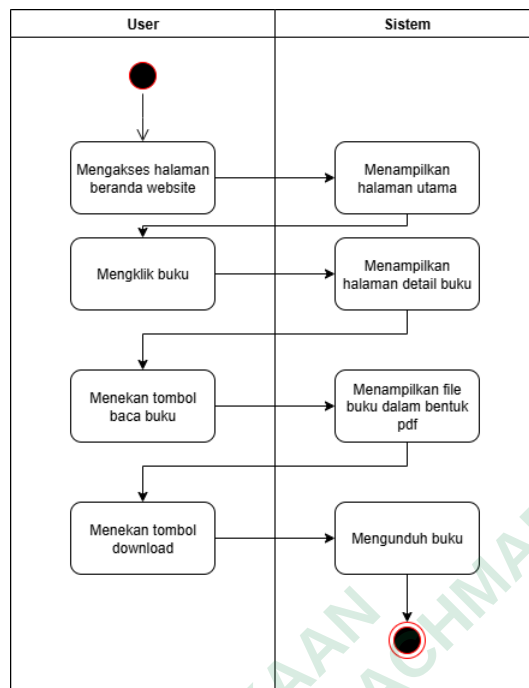
Gambar 3.10 Activity Diagram Melihat Buku Berdasarkan Kategori

Pada *activity diagram* proses melihat rekomendasi buku, alur dimulai ketika *user* berada di halaman utama. Lalu, *user* memilih salah satu buku yang tersedia, sehingga sistem memberikan tampilan halaman detail buku tersebut. Halaman ini memuat informasi penting seperti judul, penulis, kategori, dan deskripsi buku tersebut. Di bagian bawah halaman detail, sistem secara otomatis menampilkan lima rekomendasi buku serupa berdasarkan kemiripan konten atau kategori. Fitur ini bertujuan untuk membantu *user* menemukan bacaan yang relevan sesuai minatnya. *Activity diagram* untuk melihat rekomendasi buku serupa dapat dilihat pada Gambar 3.11.



Gambar 3.11 Activity Diagram Melihat Rekomendasi Buku Serupa

Pada *activity diagram* proses membaca dan mengunduh buku, alur dimulai ketika *user* berada di halaman utama. Selanjutnya, *user* memilih salah satu buku yang tersedia, sehingga sistem menampilkan halaman detail buku. Halaman ini menyajikan informasi penting seperti judul, penulis, kategori, dan deskripsi buku. Terdapat tombol “Baca Buku” yang akan mengarahkan *user* ke tampilan PDF dari file buku, yang dapat langsung dibaca secara daring maupun diunduh untuk dibaca secara luring. *Activity diagram* membaca dan mengunduh buku ditampilkan pada Gambar 3.12.



Gambar 3.12 Activity Diagram Membaca Dan Mengunduh Buku

3.4.3 Struktur Basis Data

Berikut ini adalah tabel-tabel yang dipakai di *platform* literasi berbasis *website* adalah sebagai berikut.

Tabel 3.1 Struktur Tabel Buku

Field	Tipe	Panjang	Keterangan
id	Int8	19	Primary Key
judul	Varchar	255	Not Null
penulis	Varchar	255	Not Null
genre	Varchar	50	Not Null
deskripsi	Text	-	Not Null
cover_url	Varchar	200	Null
cover_file	Varchar	100	Null
file_buku	Varchar	100	Not Null

Tabel 3.2 Struktur Tabel *User*

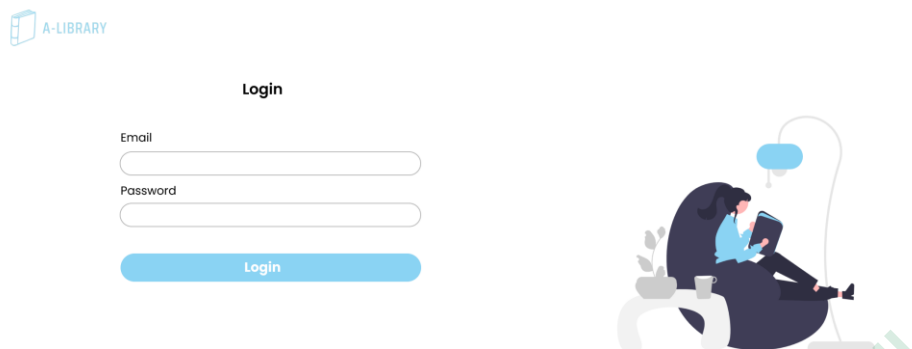
Field	Type	Panjang	Keterangan
<i>last_login</i>	Datetime	-	Not Null
<i>is_superuser</i>	Bool	-	Not Null
<i>id_user</i>	UUID	-	Primary Key
<i>name</i>	Varchar	255	Not Null
<i>email</i>	Varchar	255	Not Null
<i>password</i>	Varchar	255	Not Null
<i>roles</i>	Varchar	10	Not Null
<i>is_active</i>	Bool	-	Not Null
<i>is_staff</i>	Bool	-	Not Null
<i>date_joined</i>	Datetime	-	Not Null

3.4.4 Perancangan Antar Muka Pengguna

Desain antarmuka dirancang secara intuitif sehingga memudahkan pengguna dalam mengoperasikannya. Gambar 3.13 memperlihatkan implementasi halaman beranda yang dapat diakses baik oleh admin maupun *user*.

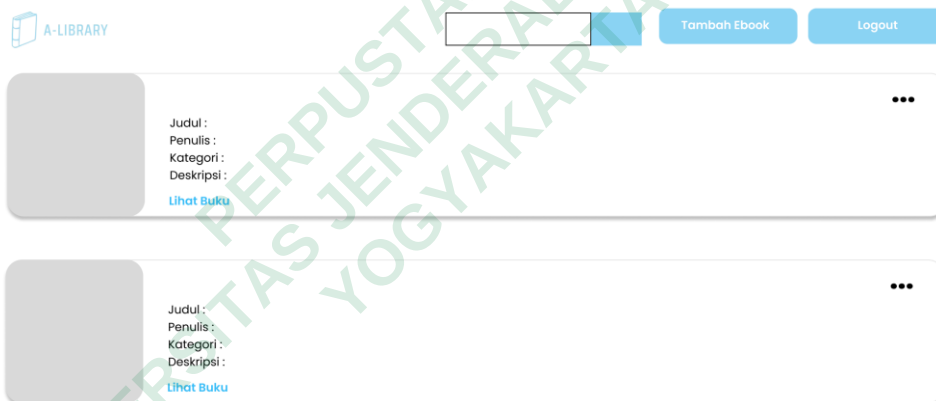
**Gambar 3.13** Halaman Beranda

Gambar 3.14 merupakan rancangan halaman *login* yang digunakan oleh admin.



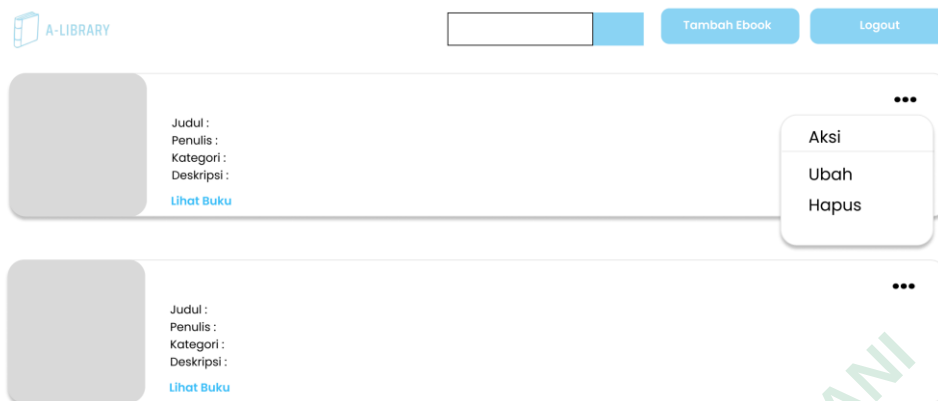
Gambar 3.14 Halaman *Login*

Gambar 3.15 ialah rancangan halaman *dashboard* admin yang terdapat tampilan data buku, fitur *search bar*, tombol tambah *ebook* dan tombol *logout*.



Gambar 3.15 Halaman *Dashboard* Admin

Gambar 3.16 merupakan rancangan halaman *dashboard* admin saat tiga titik di ujung kanan atas dari data buku telah ditekan, kemudian menampilkan pilihan ubah dan hapus yang dapat digunakan untuk mengubah atau menghapus data buku.



Gambar 3.16 Halaman Dashboard Admin tombol aksi

Gambar 3.17 dan gambar 3.18 merupakan rancangan halaman form upload buku untuk menambahkan data buku baru ke *database*.

The 'Upload Buku' form contains the following fields and elements:

- Judul**: A text input field.
- Kategori Buku**: A text input field.
- Penulis**: A text input field.
- Deskripsi**: A text input field.
- Sampul Buku**: A section containing a 'Pilih File' button and the text 'Tidak ada file yang dipilih'.

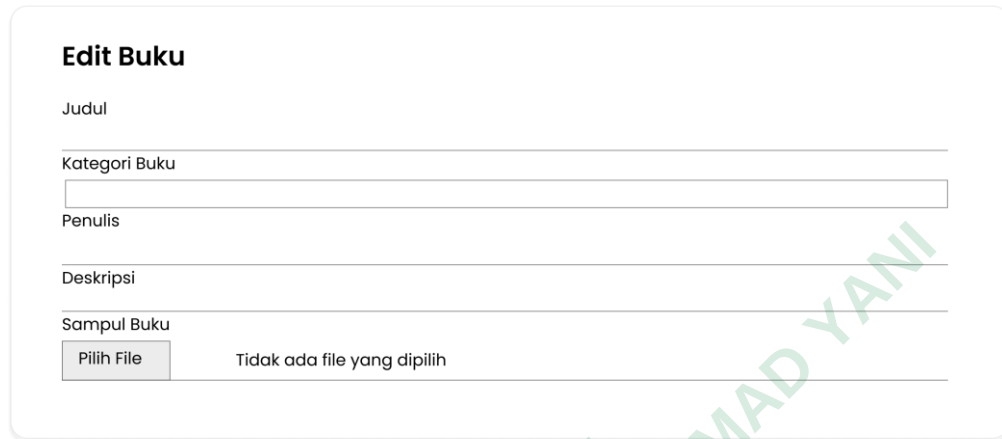
Gambar 3.17 Form Upload Buku

This form shows the file selection section for uploading a book:

- Sampul Buku**: A section with a 'Pilih File' button and the text 'Tidak ada file yang dipilih'.
- File Buku**: A section with a 'Pilih File' button and the text 'Tidak ada file yang dipilih'.
- Submit**: A blue button at the bottom.

Gambar 3.18 Form Upload Buku

Gambar 3.19 dan gambar 3.20 merupakan rancangan halaman *form* edit buku untuk memperbarui data buku ke *database*.



Edit Buku

Judul

Kategori Buku

Penulis

Deskripsi

Sampul Buku

Pilih File Tidak ada file yang dipilih

Gambar 3.19 *Form* Edit Buku



Sampul Buku

Pilih File Tidak ada file yang dipilih

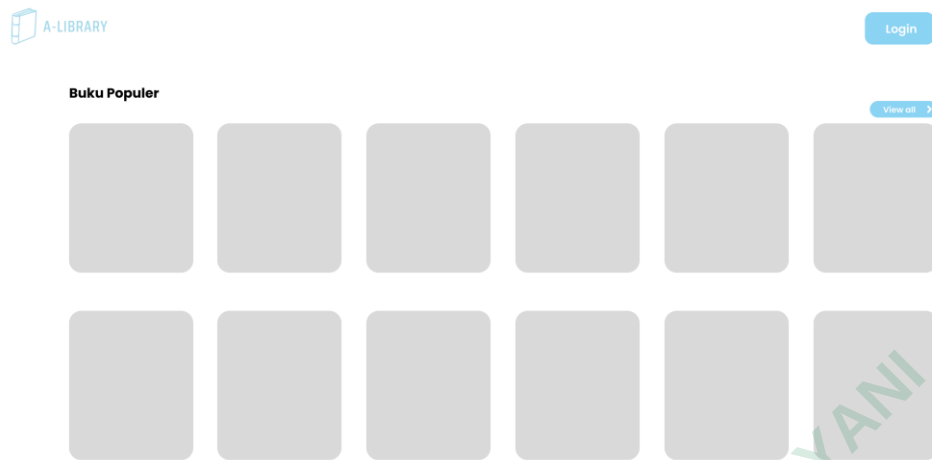
File Buku

Pilih File Tidak ada file yang dipilih

Submit

Gambar 3.20 *Form* Edit Buku

Gambar 3.21. merupakan desain halaman utama yang menampilkan beberapa buku yang populer. Selain itu, terdapat tombol login untuk admin yang ingin masuk ke dashboard admin. Di halaman ini terdapat tombol “view all” yang akan mengarah ke halaman *list* buku ketika *user* menekannya.



Gambar 3.21 Halaman Utama

Gambar 3.22. menunjukkan desain halaman daftar buku. Pada halaman ini terdapat *sidebar* di sisi kiri yang menampilkan daftar kategori buku. Selain itu, tersedia fitur pencarian yang terletak di pojok kanan atas. Daftar buku ditampilkan di bagian utama halaman, dan setiap buku dapat diklik untuk mengakses halaman detailnya.



Gambar 3.22 Halaman *List Buku*

Gambar 3.23. menampilkan rancangan halaman detail buku. Halaman ini memuat informasi lengkap mengenai buku, seperti judul, penulis, genre, dan deskripsi. Terdapat pula tombol “Baca Buku” yang memungkinkan pengguna untuk

membaca atau mengunduh buku. Di bagian bawah halaman, sistem juga menampilkan lima rekomendasi buku serupa.



Gambar 3.23 Halaman Detail Buku

3.5 PENULISAN KODE PROGRAM

Tahap ketiga dalam metode *Waterfall* adalah *development* atau pengembangan, yaitu proses mengubah desain sistem menjadi kode pemrograman yang dapat dipahami dan dijalankan oleh komputer. Pada tahap ini, sistem rekomendasi buku berbasis *content-based filtering* diterapkan ke dalam bentuk aplikasi *website* memakai bahasa pemrograman Python. Proses implementasi sistem dibagi menjadi dua bagian, yaitu *backend* dan *frontend*. Bagian *backend* meliputi pengembangan logika sistem dan pengaturan hak akses bagi admin serta *user*, sedangkan bagian *frontend* merupakan antarmuka pengguna yang dapat diakses oleh pengunjung secara umum. Untuk membangun sistem ini, digunakan *framework* Django karena mendukung pengembangan web berbasis Python secara efisien dan terstruktur.

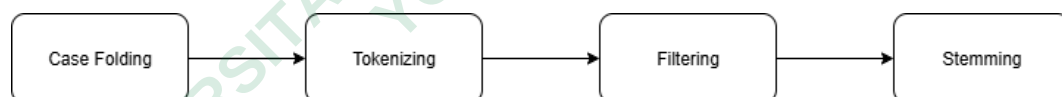
Sementara itu, implementasi basis data menggunakan PostgreSQL. Selain pengembangan web, pada tahap ini juga dilakukan pengolahan dataset berisi 500

data buku untuk menghasilkan sistem rekomendasi. Proses ini dilakukan menggunakan *library* Python seperti Pandas untuk manipulasi data dan *Scikit-Learn* untuk membangun algoritma *content-based filtering*. Dalam pengolahan dataset ini, dilakukan pembobotan nilai teks memanfaatkan metode TF-IDF (*Term Frequency-Inverse Document Frequency*) guna mengalkulasi tingkat kepentingan sebuah kata pada deskripsi buku, lalu dihitung tingkat kemiripan antar buku menggunakan *cosine similarity*. Nilai kemiripan inilah yang menjadi dasar sistem dalam merekomendasikan buku-buku yang relevan bagi pengguna berdasarkan konten deskripsi yang mirip.

3.5.1 Preprocessing

Preprocessing ialah langkah awal dalam proses klasifikasi teks yang bertujuan guna mempersiapkan data sebelum memasuki tahapan pemrosesan lebih lanjut. Tahapan ini mengubah teks mentah menjadi format yang lebih terstruktur dan bersih agar menghasilkan informasi berkualitas tinggi yang siap digunakan pada proses berikutnya (Khairunnisa et al., 2021).

Proses *preprocessing* data dilakukan melalui beberapa langkah, antara lain *case folding*, *tokenizing*, *filtering* (*stopword removal*), dan *stemming* (Zayyad, 2021). Langkah-langkah *preprocessing* data ditampilkan di Gambar 3.24.



Gambar 3.24 Tahapan *Preprocessing*

Case folding

Tahap *case folding* dijalankan dengan memodifikasi seluruh teks dalam dokumen data buku seperti judul, genre, penulis, dan deskripsi beralih menjadi huruf kecil semua. Langkah ini bertujuan guna menyamakan format penulisan setiap kata, sehingga tidak terjadi perbedaan antara kata yang sama hanya karena penggunaan huruf kapital atau tidak. Contoh penerapannya ditampilkan di tabel 3.3.

Tabel 3.3 Proses *Case folding*

Sebelum	Sesudah
Buku tentang dua jenis pola pikir (mindset) ya...	buku tentang dua jenis pola pikir (mindset) ya...
Sebuah novel karya Brian Khrisna yang mengangk...	sebuah novel karya brian khrisna yang mengangk...
Sebuah karya fiksi sejarah yang berlatar be...	sebuah karya fiksi sejarah yang berlata...
Buku tentang bagaimana perilaku manusia memeng...	buku tentang bagaimana perilaku manusia memeng...
Sebuah novel yang menceritakan kisah Sa...	sebuah novel yang menceritakan kisah sa...

Tokenizing

Tahapan *tokenizing* bertujuan untuk memecah *string input* menjadi potongan-potongan kata. Dalam proses ini, dilakukan juga *cleaning* pada data buku untuk menghilangkan simbol, angka, *backslash*, tanda baca, karakter non-ASCII, *whitespace* di awal dan akhir kalimat, URL, serta mengubah *multiple whitespace* menjadi satu, dan menghapus karakter tunggal. Dengan membersihkan data terlebih dahulu, ukuran dokumen menjadi lebih ringkas supaya proses *tokenizing* mampu dijalankan dengan lebih efisien. Contoh hasil dokumen data buku setelah tahap *tokenizing* ditampilkan di Tabel 3.4.

Tabel 3.4 Proses *Tokenizing*

Sebelum	Sesudah
buku tentang dua jenis pola pikir (mindset) ya...	[buku, tentang, dua, jenis, pola, pikir, minds...]
sebuah novel karya brian khrisna yang mengangk...	[sebuah, novel, karya, brian, khrisna, yang, m...]
sebuah karya fiksi sejarah yang berlatar...	[sebuah, karya, fiksi, sejarah, yang, berlatar, ...]
buku tentang bagaimana perilaku manusia memeng...	[buku, tentang, bagaimana, perilaku, manusia, ...]
sebuah novel yang menceritakan kisah sa...	[sebuahnovel, yang, menceritakan, kisah, santi...]

Filtering

Filtering (stopword removal) merupakan langkah untuk mengekstraksi kata-kata signifikan dari dokumen data buku. Dalam tahap ini, kata-kata yang dianggap tidak memberikan makna deskriptif, seperti kata sambung, kata depan, dan kata ganti, akan dihapus. Contohnya termasuk “di”, “yang”, “dan”, “ke”, serta “dari”. Tujuan dari proses ini ialah guna mengurangi jumlah kata dalam dokumen agar analisis menjadi lebih efisien dan terfokus pada kata-kata yang mengandung makna signifikan. Contoh hasil dari proses *filtering* ditampilkan pada Tabel 3.5.

Tabel 3.5 Proses *Filtering*

Sebelum	Sesudah
[buku, tentang, dua, jenis, pola, pikir, minds...]	[buku, jenis, pola, pikir, mindset, berbeda, f...]
[sebuah, novel, karya, brian, khrisna, yang, m...]	[sebuah, novel, karya, brian, khrisna, mengang...]
[sebuah, karya, fiksi, sejarah, yang, berlatar, ...]	[sebuah, karya, fiksi, sejarah, berlatar, belaka...]
[buku, tentang, bagaimana, perilaku, manusia, ...]	[buku, bagaimana, perilaku, manusia, memengaru...]
[sebuahnovel, yang, menceritakan, kisah, santi...]	[sebuah, novel, menceritakan, kisah, santiago, s...]

Stemming

Tahap *stemming* dilakukan untuk mengembalikan setiap kata yang telah melalui proses *filtering* ke bentuk dasarnya. Proses ini bertujuan menghapus imbuhan seperti awalan (*prefix*), akhiran (*suffix*), sisipan (*infix*), maupun kombinasi keduanya (*confix*). Dalam penelitian ini, proses *stemming* memanfaatkan *library* Sastrawi yang mampu memodifikasi kata-kata berimbuhan dalam dokumen data buku menjadi bentuk kata dasarnya. Contoh penerapan dari proses ini ditampilkan di Tabel 3.6.

Tabel 3.6 Tabel Proses *Stemming*

Sebelum	Sesudah
[buku, jenis, pola, pikir, mindset, berbeda, f...]	[buku, jenis, pola, pikir, mindset, beda, fixe...]
[sebuah, novel, karya, brian, khrisna, mengang...]	[buah, novel, karya, brian, khrisna, angkat, i...]
[sebuah, karya, fiksi, sejarah, berlatar, belaka...]	[buah, karya, fiksi, sejarah, latar, belakang, ...]
[buku, bagaimana, perilaku, manusia, memengaru...]	[buku, bagaimana, perilaku, manusia, pengaruh, ...]
[sebuah, novel, menceritakan, kisah, santiago, s...]	[buah, novel, cerita, kisah, santiago, orang, ...]

3.5.2 Perhitungan Kemiripan

Perhitungan skor kemiripan antar buku dilakukan melalui dua langkah utama. Langkah pertama memanfaatkan algoritma TF-IDF guna menentukan bobot masing-masing *term* dalam deskripsi buku. Selanjutnya, tahap kedua melibatkan kalkulasi tingkat kemiripan antar buku dengan menerapkan metode *cosine similarity*.

Pembobotan Kata TF-IDF *Vectorizer*

Pemberian bobot pada kata-kata dilaksanakan dengan memakai algoritma TF-IDF, yang mengukur tingkat kepentingan suatu kata dalam merepresentasikan konten sebuah buku. Pada metode ini, bobot kata akan meningkat seiring dengan frekuensi kemunculan kata tersebut dalam buku yang dianalisis, namun akan menurun apabila kata tersebut banyak muncul di buku lain dalam kumpulan data. TF-IDF memberi bobot nilai yang lebih besar pada kata-kata yang bersifat lebih spesifik dan relevan terhadap buku tertentu. Sebagai contoh, berikut adalah tiga deskripsi yang telah diproses melalui tahapan *preprocessing*.

- Kalimat 1: “misteri bunuh orang wanita kaya”
- Kalimat 2: “bunuh orang tumpang kereta mewah”
- Kalimat 3: “bunuh pesta mati orang pendeta sebab racun”

Berdasarkan ketiga kalimat tersebut, dilakukan perhitungan TF-IDF menggunakan rumus $idf_i = \log \left(\frac{D}{df_i} \right)$.

Keterangan:

D = dokumen ke-d

t = *term* ke-t dari dokumen

tf = jumlah kemunculan *term* i dalam dokumen

idf = *Inversed Document Frequency*

df = banyak dokumen yang mengandung term i

Dengan menggunakan perhitungan berdasarkan rumus tersebut, proses pemberian bobot nilai dapat dilakukan sebagaimana ditampilkan di Tabel 3.7.

Tabel 3.7 Tabel Pembobotan TF-IDF

No	Kata	Doc 1	Doc 2	Doc 3	df	Idf	Tf.idf		
							Doc 1	Doc 2	Doc 3
1	misteri	1	0	0	1	$\text{Log}(3/1)=0.477$	0.477	0	0
2	bunuh	1	1	1	3	$\text{Log}(3/3)=0$	0	0	0
3	orang	1	1	1	3	$\text{Log}(3/3)=0$	0	0	0
4	wanita	1	0	0	1	$\text{Log}(3/1)=0.477$	0.477	0	0
5	kaya	1	0	0	1	$\text{Log}(3/1)=0.477$	0.477	0	0
6	tumpang	0	1	0	1	$\text{Log}(3/1)=0.477$	0	0.477	0
7	kereta	0	1	0	1	$\text{Log}(3/1)=0.477$	0	0.477	0
8	mewah	0	1	0	1	$\text{Log}(3/1)=0.477$	0	0.477	0
9	pesta	0	0	1	1	$\text{Log}(3/1)=0.477$	0	0	0.477
10	pendeta	0	0	1	1	$\text{Log}(3/1)=0.477$	0	0	0.477
11	sebab	0	0	1	1	$\text{Log}(3/1)=0.477$	0	0	0.477
12	racun	0	0	1	1	$\text{Log}(3/1)=0.477$	0	0	0.477

Penerapan Algoritma *Cosine similarity*

Cosine similarity digunakan guna menghitung tingkat kesamaan antara dua vektor berdasarkan sudut di antara keduanya dalam ruang multidimensi di suatu ruang vektor. Dengan metode ini, sistem dapat memutuskan sejauh mana kesamaan antara satu buku dengan buku yang lain. Proses persamaan dilakukan tidak hanya

berdasarkan bobot kata dari hasil TF-IDF, tetapi juga mempertimbangkan sudut antara vektor-vektor dokumen tersebut.

Sebagai ilustrasi, dokumen A dan B dipakai untuk menguji performa sistem. Kedua dokumen tersebut sudah melewati tahap proses *preprocessing* sebelumnya. Jika A = “matahari terbit sebelah timur” dan B = “matahari terbenam sebelah barat atau sebelah timur”. Maka tabel frekuensinya adalah sebagai berikut.

Tabel 3.8 Tabel Frekuensi

No	Term	T(A)	T(B)
1	matahari	1	1
2	terbit	1	0
3	sebelah	1	2
4	timur	1	1
5	terbenam	0	1
6	barat	0	1
7	atau	0	1

Vektor A dan vektor B dalam Tabel 3.8 merepresentasikan masing-masing term dari dokumen A dan B untuk menunjukkan frekuensi kemunculan setiap kata unik dalam dokumen. Nilai-nilai tersebut kemudian akan digunakan dalam perhitungan *cosine similarity*.

a. Vektor A = (1,1,1,1,0,0,0)

b. Vektor B = (1,0,2,1,1,1,1)

Persamaan *cosine similarity* dapat dikalkulasikan dengan memakai persamaan di bawah ini.

$$\cos(A, B) = \frac{\sum_{i=1}^n A_i B_i}{\left| \left| \sqrt{\sum_{i=1}^n A_i^2} \right| \right| \cdot \left| \left| \sqrt{\sum_{i=1}^n B_i^2} \right| \right|}$$

$$\cos(A, B) = \frac{(1 \times 1) + (1 \times 0) + (1 \times 2) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 1)}{\sqrt{1^2 + 1^2 + 1^2 + 1^2 + 0^2 + 0^2 + 0^2} \times \sqrt{1^2 + 0^2 + 2^2 + 1^2 + 1^2 + 1^2 + 1^2}}$$

$$\cos(A, B) = \frac{1 + 0 + 2 + 1 + 0 + 0 + 0}{\sqrt{1 + 1 + 1 + 1 + 0 + 0 + 0} \times \sqrt{1 + 0 + 4 + 1 + 1 + 1 + 1}}$$

$$\text{Cos (A, B)} = \frac{4}{\sqrt{4} \times \sqrt{9}}$$

$$\text{Cos (A, B)} = \frac{4}{2 \times 3} = \frac{4}{6} = 0.6667$$

3.6 PENGUJIAN SISTEM

Tahapan keempat dalam metode *Waterfall* adalah pengujian sistem (*testing*). Dalam tahap ini, sistem yang telah dibuat akan dilaksanakan uji coba guna mengonfirmasi bahwa seluruh fitur berjalan sesuai dengan fungsinya dan sistem rekomendasi buku bisa menampilkan hasil yang relevan. Pengujian dijalankan dengan metode *Black box Testing*, yaitu teknik pengujian perangkat lunak yang berpusat pada pengujian fungsi-fungsi sistem terlepas dari struktur internal atau kode program yang mendasarinya. Tujuan dari pengujian ini ialah guna memastikan bahwa *input* yang dimasukkan dapat memberikan hasil *output* yang sesuai dengan yang didambakan, serta memastikan bahwa antarmuka pengguna bekerja dengan baik.

Dengan pendekatan ini, pengujian difokuskan pada fungsionalitas sistem, termasuk pengecekan apakah fitur-fitur seperti pencarian buku, tampilan detail buku, pengunduhan file, dan sistem rekomendasi yang berbasis *content-based filtering* berjalan secara optimal. Selain itu, pengujian juga dilaksanakan untuk mengevaluasi apakah hasil rekomendasi yang ditampilkan sesuai dengan deskripsi dan preferensi yang diharapkan oleh *user*.